

# OpenBiblio Documentation

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>About this documentation</b>	<b>1</b>
2.1	Licensing	1
2.2	Contributors	1
<b>3</b>	<b>The OpenBiblio Community</b>	<b>1</b>
3.1	Getting help	1
3.2	Reporting bugs	1
3.3	Contributing code	1
3.4	Contributing documentation	1
<b>4</b>	<b>Installing OpenBiblio</b>	<b>2</b>
4.1	System requirements	2
4.2	Installing OpenBiblio 1.0 on a Windows computer	2
4.2.1	Install prerequisite software	2
4.2.2	Install OpenBiblio	3
4.2.3	Using the OpenBiblio installer.	3
4.3	Installing OpenBiblio 1.0 on a shared Linux host	3
4.4	Installing OpenBiblio 1.0 on a Linux host with full access	3
4.4.1	Install prerequisite software	3
4.4.2	Install yaz (optional)	4
4.4.3	Start the Apache and MySQL services	4
4.4.4	Create a MySQL database	4
4.4.5	Prepare OpenBiblio for its installation	4
4.4.6	Using the OpenBiblio installer.	5
<b>5</b>	<b>OpenBiblio System Administration</b>	<b>5</b>
5.1	Managing your libraries	6
5.1.1	Adding a site	6
5.1.2	Deleting a site	6
5.1.3	Library-specific settings	6
5.2	Managing your users	6
5.2.1	Working with passwords	7
5.2.2	Deleting users	7
5.3	Managing your catalog	7
5.3.1	Managing media types	7
5.3.2	Managing collections	8

---

Circulating collections . . . . .	8
5.3.3 Managing displayed fields . . . . .	8
Adding and re-arranging displayed fields . . . . .	9
Removing and renaming displayed fields . . . . .	9
5.3.4 Managing custom fields for copies . . . . .	9
5.3.5 Managing copy catalog options . . . . .	9
Basic configuration . . . . .	9
Server-by-server configuration . . . . .	10
5.4 Configuring the OPAC . . . . .	10
5.5 Managing locales . . . . .	10
5.5.1 Character set . . . . .	10
5.5.2 Creating a new locale . . . . .	10
Create a directory for your new locale . . . . .	10
5.6 Managing the MySQL database . . . . .	11
5.7 Managing the server . . . . .	11
5.7.1 Logs . . . . .	11
5.7.2 Not yet implemented features . . . . .	11
OPAC URL . . . . .	11
Requests . . . . .	11
<b>6 Circulation</b> . . . . .	<b>11</b>
6.1 Checking out materials . . . . .	12
6.1.1 Retrieving a member's account. . . . .	12
6.1.2 Checking out an item . . . . .	12
6.2 Checking in materials . . . . .	12
6.3 Showing a member's circulation history . . . . .	12
6.3.1 Retrieving a member's account. . . . .	13
6.3.2 Viewing the history . . . . .	13
6.4 Creating a new member . . . . .	13
6.5 Holds . . . . .	13
6.6 Bookings . . . . .	13
<b>7 Cataloging</b> . . . . .	<b>13</b>
7.1 Orientation to OpenBiblio cataloging . . . . .	13
7.1.1 Call numbers . . . . .	14
7.1.2 Special subfields . . . . .	14
7.2 Copy cataloging . . . . .	14
7.2.1 Adding an item . . . . .	14
7.3 Original cataloging . . . . .	14

---

7.3.1	Adding an item	15
7.4	Adding cover images to records	15
7.5	Importing MARC records	15
7.6	Importing CSV records	15
7.6.1	The CSV File	15
7.6.2	Importing the CSV file	16
7.7	Bulk delete records	16
<b>8</b>	<b>Research</b>	<b>16</b>
8.1	Local search	16
8.2	DOI search	16
8.3	Cover Photos	17
8.4	Cart	17
<b>9</b>	<b>Reports</b>	<b>17</b>
9.1	Running a report	17
9.1.1	Exporting to different formats	17
9.2	Adding new reports	17
<b>10</b>	<b>Managing OpenBiblio Plugins</b>	<b>17</b>
10.1	Enabling plugins	17
10.2	Plugins bundled with OpenBiblio	18
10.2.1	CSS Utilities (cssUtils)	18
	WARNINGS	18
10.2.2	Pull-Down List Manager (List Manager)	18
10.2.3	Media Fields	18
10.2.4	Orphan File finder (orpahnFiles)	18
10.2.5	Translation Utilities (transUtils)	19
10.2.6	Call Number Utilities (CallNoUtils)	19
	Find Records without call numbers	19
	Dry Run of call number add	19
	Add call numbers to records	19
10.2.7	Online photos (OpenLibrary)	19
10.3	Creating a new plugin	19
10.3.1	Design the plugin	19
	Create a plugin form	20
	Create a plugin JS file	20
	Create a plugin server	21
10.3.2	Add your plugin to the OpenBiblio menu	21

---

<b>11 OpenBiblio Recipes</b>	<b>22</b>
11.1 A successful initial configuration . . . . .	22
11.2 Adding a link to your system documentation . . . . .	22
11.3 Changing the timeout length on HMAC requests . . . . .	23
<b>12 Accessibility</b>	<b>23</b>
12.1 Skipping navigation menus . . . . .	23

## Introduction

This manual provides documentation for users of OpenBiblio 1.0, an automated library system.

OpenBiblio 1.0 has a public OPAC interface and a staff interface. You can successfully use either interface on your computer, tablet, or phone.

## About this documentation

This documentation uses ASCIIDoc, using a workflow borrowed from the Evergreen ILS Documentation Interest Group.

## Licensing

This work is licensed under a [Creative Commons Attribution 4.0 License](#).

## Contributors

- Megan Bell
- Fred LaPlante
- Jane Sandberg

## The OpenBiblio Community

We are happy to welcome you to the OpenBiblio Community! You have joined a worldwide group of librarians, developers, community centers.

## Getting help

You can get help from the mailing list or the sourceforge forums.

## Reporting bugs

You can report bugs in the bitbucket repository.

## Contributing code

You can contribute custom plugins, themes, and reports!

If you want to create a new feature for the base OpenBiblio software, start out by proposing your idea on the bitbucket issues interface. If your proposal makes sense and you get some consensus with the other developers, you can contribute your code, test it, and contribute it back to the main repository as a pull request.

## Contributing documentation

You can find instructions at [https://github.com/openbiblio/openbiblio\\_docs](https://github.com/openbiblio/openbiblio_docs).

---

## Installing OpenBiblio

### System requirements

We have tested OpenBiblio installation on Windows 7 and Ubuntu Linux 14.04.

We also plan to test other versions of Windows, Ubuntu Linux 12.04, Debian Linux, Fedora Linux, Mac OSX, and CentOS Linux.

To install OpenBiblio, you will need PHP version 5.4 or higher. OpenBiblio is fully compatible with PHP 7.

You will need MySQL. OpenBiblio has been tested with versions 5.5 and 5.7.

You will also need some Web server program. OpenBiblio has been tested with both Apache and nginx. Web server software.

### Installing OpenBiblio 1.0 on a Windows computer

You can turn your personal Windows computer into an OpenBiblio server. You will need administrator access to your computer, an Internet connection, and you may also need to check with your local IT experts that you are not behind a restrictive firewall.

Note that turning your personal computer into a server carries some risks, especially if your server is broadcasting across the web instead of just a local network. We recommend that you run this software on a dedicated computer that does not contain any sensitive or private data.

### Install prerequisite software

1. Download EasyPHP's Webserver from <http://easypHP.org>. We have tested these instructions with version 14.1 of EasyPHP.
  2. Agree to the relevant licenses and choose an installation folder that is convenient for you.
  3. Once EasyPHP is installed, it will appear as an icon in your system tray. On Windows 7, click on this icon and open the dashboard.
  4. The dashboard will open in your default Web browser. Once it does, click Settings in the top right corner of the screen.
  5. Click on HTTP SERVER. Install and start this service.
  6. Click on DB SERVER. Install and start this service.
  7. Go back to the dashboard.
  8. Under modules, you should now see an option to administer MySQL through a program called PhpMyAdmin.
  9. Open PhpMyAdmin. The default username is root and there is no default password. Click on Users.
  10. Click on "Edit Privileges" next to root.
  11. Click on "Change Password".
  12. Click on Add user.
  13. Choose a memorable username, like *obiblio*, and enter it. We suggest using a name that only contains letters, numbers, and underscore characters.
  14. In the host field, enter "localhost".
  15. Create a long, secure, memorable password.
  16. Under "Database for user", there should be a checkbox for "Create database with same name and grant all privileges". Check this box.
  17. Click go.
-

## Install OpenBiblio

1. Open the EasyPHP dashboard.
2. Go to Settings and open the HTTP SERVER settings.
3. Note the Document Root field. This is the folder on your computer where OpenBiblio will be running.
4. Open the DB SERVER settings. Note the parameters area.
5. Download OpenBiblio and place the files into the Document Root folder.
6. Go back to the EasyPHP dashboard and open the HTTP SERVER settings once more.
7. There is a link there to open up OpenBiblio under the URL field. Click on it. The installer should appear.

## Using the OpenBiblio installer.

1. Fill out the required information using the parameters of your MySQL setup. The hostname should be the IP address and port in the format 192.168.333.333:3388, where 192.168.333.333 is the IP address and 3388 is port number.
2. Install test data if you would like.
3. Log in to your new OpenBiblio system. The default user is admin with password admin.

## Installing OpenBiblio 1.0 on a shared Linux host

These instructions will help you set up OpenBiblio on a Web server provided by Dreamhost, Hostgator, or a similar service.

1. Create a database using your hosting service's interface. Note the host, username, password, and database name.
2. Download the Filezilla FTP program

## Installing OpenBiblio 1.0 on a Linux host with full access

These instructions consist mainly of commands that you can run in your terminal to quickly get a working installation of OpenBiblio. If you are not comfortable with using the terminal, feel free to use the instructions above for a shared Linux host.

These procedures have been tested on Fedora 21 and Ubuntu 14.04.

## Install prerequisite software

OpenBiblio requires apache, PHP, and MySQL. If you intend to copy catalog records from other libraries, we strongly recommend that you use yaz. The following steps include a YAZ install.

On Ubuntu 14.04:

1. `sudo apt-get install apache2 php5 mysql-server yaz php5-dev php5-mysql php-pear libyaz4-  
dev unzip`
2. use your favorite editor to add `"extension=yaz.so"` to `/etc/php/apache/php.ini`  
TIP: search for `'extension='` to find the right location
3. `sudo pecl install yaz`

---

### Tip

This will prompt you to choose a root password for MySQL. Make sure it is a secure one!

---



On Fedora 21:

```
sudo yum install httpd php mysql-server php-devel php-mysql php-pear unzip yaz libyaz-devel ↵  
gcc  
sudo mysql_secure_installation
```

---

**Tip**

mysql\_secure\_installation will ask you for a current root password, which you may leave blank. Be sure to give root a new, secure password. You should also choose to remove anonymous users, disallow root login remotely, and remove the test database.

---

### Install yaz (optional)

Next, install yaz as a PHP extension using pecl.

```
sudo pecl install yaz
```

Add a new line to your php.ini file. This file can be found at /etc/php/php.ini in Fedora and /etc/php5/apache/php.ini in Ubuntu. The line should be something like the following:

```
extension=yaz.so
```

### Start the Apache and MySQL services

On Fedora 21:

```
sudo service httpd start  
sudo service mysqld start
```

On Ubuntu:

```
sudo service apache2 start  
sudo service mysql start
```

### Create a MySQL database

First, open up MySQL.

```
mysql -uroot -p #MySQL will ask you for the root password you just created
```

Next, type the following commands to create a user and database for OpenBiblio to use.

```
CREATE DATABASE obiblio;  
GRANT ALL PRIVILEGES ON obiblio.* TO obib_user@localhost IDENTIFIED BY 'obib_password';
```

### Prepare OpenBiblio for its installation

```
wget -O obib.zip https://bitbucket.org/mstetson/obiblio-1.0-wip/get/26f88507b580.zip  
unzip obib.zip
```

Delete the default "It Works" page from /var/www/html.

```
sudo mv ./* /var/www/html/
```

On Fedora 21, you may need to run the following command to configure SELinux to allow access to the folder.

```
restorecon -r /var/www/html
```

---

**Using the OpenBiblio installer.**

1. Fill out the required information using the parameters of your MySQL setup. The hostname should be the IP address and port in the format 192.168.333.333:3388, where 192.168.333.333 is the IP address and 3388 is port number.
2. Install test data if you would like.
3. Log in to your new OpenBiblio system. The default user is admin with password admin.

To protect sensitive member data, we strongly encourage you to set up HTTPS, so that anybody who intercepts OpenBiblio data will see an encrypted jumble, rather than your members' information.

The following is extracted from "Ubuntu Unleashed", 2016 edition, Sams, pp 513 - 515.

All below assumes you are using Apache2 running on a Ubuntu server.

1. Enable the following module: `sudo a2enmod ssl`

```
to configure Apache2 for HTTPS using the default configuration for testing:
sudo a2ensite default-ssl
and restart using
sudo service apache2 restart
```

2. create a self-signed certificate and key first generate a key `openssl genrsa -des3 -out server.key 2048`

```
next a signing request (CSR)
```

```
openssl req -new -key server.key -out server-csr
```

```
generate a certificate for 1 year (365 days) larger numbers are allowed - I ↔
use 36500!
```

```
openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
```

```
copy to the certificate to a 'proper' location
```

```
sudo cp server.crt /etc/ssl/certs/
```

```
copy the key to a proper location
```

```
sudo cp server.key /etc/ssl/private/
```

```
edit apache2's default -ssl file /etc/apache2/sites-available/default-ssl to ↔
read as
```

```
SSLEngine on
```

```
SSLCertificateFile /etc/ssl/certs/server.crt
```

```
SSLCertificateKeyFile /etc/ssl/private/server.key
```

```
to configure Apache2 to use this info
```

```
sudo a2ensite default-ssl
```

Now restart Apache; be prepared to input the certificate's key password.

Be sure to save this password/phrase, you will need it every time you restart ↔  
Apache.

```
sudo service apache2 restart
```

**OpenBiblio System Administration**

Thanks for being an OpenBiblio system administrator. We're glad that you are helping your library in this way.

## Managing your libraries

OpenBiblio 1.0 introduced the ability to manage multiple libraries with a single OpenBiblio installation. Each library is considered a separate "site," with its own name, collection, address, hours, closed dates, and other settings.

### Adding a site

1. Go to Admin → Sites.
2. Click "Add New".
3. Fill out the relevant information. Choose the calendar that includes the dates your new library will be closed. See below for more information about the calendars.

**Warning**

If you do not choose a calendar, the new site will not be able to calculate due dates properly and you will not be able to check out any copies at this site.

---

4. Click "Add."

### Deleting a site

1. Ensure that there are no copies at the site that you'd like to delete.
2. Sign in as a different site.
3. Go to Admin → Sites.
4. Delete the site.

---

**Note**

OpenBiblio does not allow you to delete the site that you are currently signed in as.

---

---

**Note**

OpenBiblio checks to make sure that there are no copies attached to the site you wish to delete. If there are still copies attached, you will have to delete them or move them to a different site before you can delete the site.

---

### Library-specific settings

If you'd like the staff view of OpenBiblio to include your library's hours and phone number, go to Admin → Library Settings → Library. Check the "Show Library Info on Staff Pages" box.

## Managing your users

To add a new staff member, go to Admin → Staff Admin. Click on the "Add New" button, found both at the top and bottom of this screen. You can then assign a username, password, name, and permission set to your staff member.

---

**Tip**

The maximum length for usernames is 20 characters. The maximum length for passwords is 32 characters.

---

**Warning**

You must assign at least one permission, or the staff member will not be able to log in.

---

### Working with passwords

Openbiblio's password management system is very straightforward. Users with Admin permission are able to change their own password or the password of any other staff account (including other administrators). Users without the Admin permission may not change anybody's password, not even their own.

To change the password of a staff account, go to Admin → Staff Admin. Click the pwd button next to the account you are working with. Enter the new password. The maximum length for passwords is 32 characters.

### Deleting users

Anybody with the admin permission may delete a user.

1. Go to Admin → Staff Admin.
2. Click the edit button next to the account you are working with.
3. Verify that you are deleting the correct user.
4. Click the delete button.

**Warning**

You may delete your account if you have the admin permission. Please be very careful when deleting staff members, particularly your own account.

---

## Managing your catalog

This section describes settings in the Admin menu related to cataloging.

### Managing media types

You can access media types in Admin → Media Types.

There are several settings you can choose for any Media Type in your catalog, which are listed below.

- *Description* - a human-readable name for what type of material you are describing.
- *Checkout Limit* - the maximum number of this type of item a member may check out.
- *Default* - If this is set to *Y*, the cataloging module will default to the current media type.

---

**Note**

You should be careful to only set one media type to be default, or else the cataloging module will run into errors

---

- *Search display lines* - This setting governs how long search results views will be. If you choose 5, for example, the search results page will display the first 5 MARC fields for items of this media type (based on the order you define in the Admin → Biblio fields interface).
  - *Image file* - You may upload your own icon image for this media type here.
-

## Managing collections

You can access media types in Admin → Collections.

There are several settings you can choose for any Collection in your catalog, which are listed below.

- *Description* - a human-readable name for your collection, such as "Fiction," "Children's books," or "Genealogy Room." We recommend that each collection correspond to a physical area in your library.
- *Item count* - this field is calculated automatically by OpenBiblio based on how many items exist in this collection.
- *Collection type* - this field allows you to decide whether items in this collection should circulate or be distributed.
- *Default* - If this is set to *Y*, the cataloging module will default to the current collection.

---

### Note

You should be careful to only set one collection to be default, or else the cataloging module will run into errors.

---

## Circulating collections

Circulating collections have a number of other settings, usually associated with a specific *due date calculator*.

- *Due date calculator* is a specific procedure to determine when items in your collection should be due back to the library.
  - *simple*: this due date calculator will simply add the requested number of days and minutes to the current date and time. If the library's calendar indicates that the library will be closed on that day, it will extend the due date to 11:59 p.m. on the next available open day.
  - *at midnight*: this calculator works very similarly to the *simple* calculator, except that it *always* extends due dates to 11:59 p.m. This calculator functions very similarly to the circulation module in OpenBiblio 0.7.x, and is the best choice for collections that you'd like to circulate for a certain number of days.
  - *before we close*: this calculator is for items that should return to the library at closing time. Items will be due after the specified number of minutes or at closing time, whichever comes first. If you specify the Pre-closing padding, items will be due the specified number of minutes before the library closes, so that members don't try to return the items as you are trying to close the library.
  - *ask me*: anytime an item in this collection circulates, the staff member will manually select the due date and time. This can be useful for interlibrary loan collections, where due dates are rarely standardized.
- *Days due back* - the length of the loan period in days for items from this collection.
- *Minutes due back* - the length of the loan period in minutes for items from this collection. You may combine this with the *days due back* field, if you would like (e.g. items can be checked out for 3 days and 45 minutes).
- *Daily late fee* - The overdue fee for each day an item is held overdue.
- *Important date* - a specific date that OpenBiblio should take into account when calculating a due date and time. This might be the end of a term at school.
- *Important date purpose* - what OpenBiblio should do with the important date.
  - *ceiling date* - OpenBiblio will not calculate any dates that fall after the important date.
  - *specific date* - OpenBiblio will always use the important date as the due date.

## Managing displayed fields

Each type of media displays certain MARC fields, and can be labeled in specific ways. For example, the field 300\$a contains page numbers for book records and number of discs for DVDs. Therefore, you might want to label these in different ways to create the clearest experience for your members and staff.

---

### Adding and re-arranging displayed fields

1. Go to Admin → Media Types. Make sure that you have the appropriate media types listed.
2. Go to Admin → Biblio fields.
3. Choose the media type you are interested in.
4. Click "Add to / Modify Layout".
5. To add new fields, use the dropdown menus in the right column to choose the field that you would like. You may include the same field multiple times if you wish.
6. To re-arrange fields, drag and drop the entries in the right column until you are satisfied.
7. When everything looks good, click "Save Layout".

### Removing and renaming displayed fields

1. Go to Admin → Biblio fields.
2. Choose the media type you are interested in.
3. Click the "Edit" button next to the field you wish to remove or rename.
4. Here you can change the "label" that displays to members and staff, as well as data standards for this field.

### Managing custom fields for copies

OpenBiblio offers some great flexibility for describing your physical items. In fact, you can add any field you'd like by going to Admin → Biblio Copy Fields.

Any fields you enter here will be displayed to your staff members as optional fields.

### Managing copy catalog options

OpenBiblio's lookup feature saves catalogers a lot of time and hassle by importing pre-existing biblio records from other libraries using the SRU or z39.50 protocols.

You must complete these configurations before using the lookup features.

### Basic configuration

To access these options, go to Admin → Online Options.

1. *Online protocol*: If you have installed yaz correctly, we recommend that you select YAZ from this menu. This will use the z39.50 protocol, which is more widely used than the SRU protocol.
  2. *Maximum hits*: the number of hits to display to catalogers.
  3. *Timeout*: the amount of time OpenBiblio should wait for a response from the lookup host.
  4. *Keep dashes*: This setting determines whether or not you would like to keep hyphen characters in ISBNs that your catalogers enter. We recommend that you set this to false, because most z39.50 and SRU servers index ISBNs without hyphens.
  5. *Call number type*: the call number scheme that you'd like to use. OpenBiblio will then attempt to retrieve a call number of that type from the remote server to add to a 099 field.
  6. Not sure what Auto Dewey does
-

7. *Default Dewey* provides a default call number for items you catalog using the Dewey Decimal Classification, in case the remote server does not provide useful Dewey Call Number information.
8. *Auto Cutter* will automatically generate a Cutter number for you.
9. *Cutter Type* asks you to choose between LoC cutters and Cutter-Sanborn cutters.
10. *Noise words* are common words that you would like to remove from your queries to get better results.
11. *Auto collection* will automatically put new copies in your default collection.
12. *Fiction name* will be the initial part of any call number for fiction books, which typically do not use formal call numbers.

### Server-by-server configuration

You will also need to configure each remote server in Admin → Online Hosts.

## Configuring the OPAC

This section will describe settings found in the admin menu.

## Managing locales

OpenBiblio can be translated into any language that you wish!

### Character set

You can use any character set that you find by running the MySQL command **SHOW CHARACTER SET**. Do this by going to Admin → Locale and entering your preferred character set. If you enter an invalid charset or leave this field blank, it will default to the *utf8* character set.

---

#### Note

The character set you specify will appear in the `<meta charset>` attribute and apply to any database calls.

---

### Creating a new locale

This requires access to the files and a good UTF-8 text editor.

#### Create a directory for your new locale

1. Create a new directory in the locale directory with the 2-character ISO 639-1 code for the language. For example, if you would like to do an Amharic translation, your directory would be called `am`.

---

#### Note

OpenBiblio traditionally uses the 2-character ISO 639-1 code, but you may also use a 3-character scheme or custom string if your language isn't represented in ISO 639-1.

---

2. Create a PHP file called `metadata.php`. Make sure that the class name begins with the ISO 639-1 code for your language.
-

```
<?php
class amMetaData {
    public function __construct() {
        $this->locale_description = "Amharic";
    }
    function pluralForm($n) {
        if ($n == 1 or $n == -1) {
            return 'singular';
        } else {
            return 'plural';
        }
    }
    function moneyFormat($amount) {
        if ($amount < 0) {
            return sprintf("(-%.br)", abs($amount));
        } else {
            return sprintf("%.br", $amount);
        }
    }
}
```

3. Add translations to trans.php.

## Managing the MySQL database

This section will describe the database

## Managing the server

This section will discuss other server-level considerations.

## Logs

This section will mention logs of particular interest to OpenBiblio administrators.

## Not yet implemented features

### OPAC URL

In Admin → Library Settings → Miscellaneous, you can set the OPAC URL. However, this is currently not used anywhere in OpenBiblio.

### Requests

In Admin → Library Settings → Requests, you can set the default from and to email addresses, as well as the default subject for emailed requests.

## Circulation

The circulation module is the heart of the OpenBiblio system.

---



## Checking out materials

To check out materials, you must first open a member's account.

### Retrieving a member's account.

1. Click on Circulation on the left side of the screen.
2. If you have member barcodes turned on, you may scan their barcode.
3. Otherwise, enter a portion of the member's last name.
4. Click on the patron you wish to retrieve. If a patron goes by a name other than their legal name, both forms of their name will display in the list.

---

#### Tip

Anybody with administrator privileges can turn member barcodes on using Tools → System Settings → Use Member Barcodes.

---

### Checking out an item

1. Type or scan the item barcode into the Check Out field.
2. Any items you have checked out will appear on the screen, provided there are no errors in the patron or copy records.
3. If there are errors (often a mis-scanned barcode), they will appear at the bottom of the screen.

## Checking in materials

1. Click on Circulation on the left side of the screen.
2. Click Check In.
3. Type or scan the barcode number.
4. If the book was checked out, it will be added to the shelving cart, which is represented by a list at the bottom of the screen.

---

#### Note

Check in only works on items that are checked out. It will not remove other statuses, such as "lost".

---

After you check in your materials, you will probably want to empty the shelving cart as well.

Does the screen display fines if the item being checked in is overdue?

## Showing a member's circulation history

OpenBiblio stores information about copies that a member has checked out previously. You can access this list after you retrieve a member account.

---

**Retrieving a member's account.**

1. Click on Circulation on the left side of the screen.
2. If you have member barcodes turned on, you may scan their barcode.
3. Otherwise, enter a portion of the member's last name.
4. Click on the patron you wish to retrieve. If a patron goes by a name other than their legal name, both forms of their name will display in the list.

**Viewing the history**

Click on the History button, near the top of the screen.

---

**Note**

Unlike some ILSs, OpenBiblio does not anonymize circulations after a certain period of time has elapsed. Please know that this can be considered a member confidentiality issue in some jurisdictions.

---

**Creating a new member**

1. Click on Members, under the circulation menu.
2. Click "Add new member".
3. Enter the desired information.
4. Some people may use a name other than their legal name. If this is the case, enter the name they prefer in the "First name" and "Last name" fields, and if it is necessary, enter their legal name into the "Legal first name" and/or "Legal last name" fields. This will allow your library to retrieve information about this member using both versions of their name.

**Holds**

You can place holds on items so you can check them out at a later time.

**Bookings**

You can also book certain items

**Cataloging**

OpenBiblio offers several ways to get your materials into your catalog and to manage the records you have entered.

**Orientation to OpenBiblio cataloging**

OpenBiblio uses the MARC format as the basis for the catalog, but its editing interfaces display human-readable field labels, rather than specific numeric MARC tags.

OpenBiblio's use of MARC is pretty standard, but there are a few specific ways it treats certain fields.

---

## Call numbers

OpenBiblio's Call Number searches search only 099\$a, which is the local call number field. However, if you'd like to add call numbers from other fields, you can enable the Call Number Utilities plugin, which allows you to populate the 099\$a fields with data from other call number fields to make them searchable. Instructions for using this plugin can be found in the OpenBiblio Plugin manual.

## Special subfields

Three subfields display somewhat differently than usual.

- Field 024 \$a (identifier) is assumed to be a DOI, so it displays as a link to that digital object.
- Field 505 \$a (contents) is typically very long, so it displays as a scrollable text field.
- Field 856 \$u (URL - included by default in the ebook and web site material types) displays as a hyperlink.

## Copy cataloging

1. Click Cataloging.
2. Click New Item.
3. Type or scan an ISBN. If you are unable to find any results using an ISBN, you can use the dropdown menus to try other search criteria.
4. If it finds a catalog record from a different library, it will fill in all the fields that it can, based on the media type and information from the other library.
5. Make sure that the call number is correct.  
If you are seeing the wrong type of call number (e.g. you see a LoC call number when your library uses Dewey Decimal), somebody with admin privileges will need to change the Admin → Online Options settings.
6. Click Submit.

## Adding an item

After you add your record, you will be able to attach an actual, physical item to it.

If your library is using barcodes, you will be able to scan or type your barcode into the barcode number field. If you have the Auto Barcode option selected, OpenBiblio will generate a barcode for you. If you are not using barcodes, OpenBiblio will generate one that will only be used for internal purposes.

If item Photos are enabled, the user will be given to capture an image of the item (usually the dust cover). This feature requires a webcam or equiv. Enabling this feature will also cause the user to be occasionally *bugged* to formally allow the browser to take photos - this is a HTML5 *feature* designed to ensure user privacy. Once photos are available, they will show up on various reports, biblio lists, etc.

## Original cataloging

You may not be able to find a record from another library. To do so, go to Cataloging → New Item and click the *Manual Entry* button.

1. Select the appropriate media type. The form will then show the fields appropriate to that media type.
-

---

**Note**

Anybody with the Admin permission may add, remove, or modify which fields are shown in this screen by going to Admin → Biblio Fields.

---

2. Be sure to fill in any fields marked with the \* character. These fields must be filled in before you finish your record.
3. If you would like to edit your record further before displaying it to your patrons, unselect the *Show in OPAC* checkbox. With this unselected, members will not be able to view your record. You may change this setting at a later time.

### Adding an item

After you add your record, you will be able to attach an actual, physical item to it.

If your library is using barcodes, you will be able to scan or type your barcode into the barcode number field. If you have the Auto Barcode option selected, OpenBiblio will generate a barcode for you. If you are not using barcodes, OpenBiblio will generate one that will only be used for internal purposes.

If item Photos are enabled, the user will be given to capture an image of the item (usually the dust cover). This feature requires a webcam or equiv. Enabling this feature will also cause the user to be occasionally *bugged* to formally allow the browser to take photos - this is a HTML5 *feature* designed to ensure user privacy. Once photos are available, they will show up on various reports, biblio lists, etc.

### Adding cover images to records

You may add cover images to your records. You may do this by uploading images from your computer or taking pictures using a webcam or other camera connected to your computer.

If you turn on the OpenLibrary plugin, you may also use images hosted on OpenLibrary's remote servers. This will spare your catalogers the effort of taking or uploading their own images.

### Importing MARC records

By default, the MARC record importer imports items as a "test load," so you can verify that there are no errors. You can set "test load" to false once you are satisfied with your settings and file.

---

**Tip**

MARC records can use any one of a number of different encoding schemes. Before you import any MARC records, be sure that your administrator has selected the correct character set in Admin → Character Set.

---

### Importing CSV records

#### The CSV File

CSV files are laid out in columns and rows. The first row must contain the heading information.

Some common headings are:

Column header	Meaning
barCo	Barcode Number
media	Media Type
coll	Collection
099\$a	Call Number
100\$a	Author
245\$a	Title

---

Column header	Meaning
245\$h	Medium
306\$a	Playing Time
260\$c	Date of Publication, Distribution, etc.

To find more headings click on the "Admin" link on the left side of the page. From there click on the link that says "Biblio Fields". There you will find different options to use for headers. The data will go from the second row down in the CSV file. The data needs to be input in the same order as the column headings in the first row. The Media Type heading requires specific data to be input in the column. The data options available for the media column are:

- audio tapes
- book
- cd audio
- cd computer
- equipment
- magazines
- maps
- video/dvd

### Importing the CSV file

Click on the "Browse" button, from there you should be prompted to open your CSV file. In the options box you will need to find the "Show all records" drop down. This drop down needs to be set as yes or you will be unable to view the records.

### Bulk delete records

This powerful tool will delete any copy barcodes you specify from the database. You may also choose to delete the bibliographic data relevant to a parent item once you have deleted the last copy on an item.



#### Warning

This action cannot be undone.

---

## Research

This menu area is intended for patrons or staff who do not have access to either the cataloging or circulation facilities.

### Local search

This menu implements an on-line card-catalog for the library.

### DOI search

A DOI (digital object identifier) can refer to an article, journal, digitized archival object, or any other digital object that may help you in your research. If you have one of these identifiers, you can paste it into the DOI search box, and OpenBiblio will open the relevant resource in a new tab for you.

---

## Cover Photos

A viewer of all library material for which photos are available. The format of the pages displayed are set using the Cover Photo tab of Library Settings under the Admin Menu. Clicking the title of the item shown will bring up the usual descriptive information. Nothing may be edited from these pages.

## Cart

Provides a list of material awaiting shelving. Again, clicking on an item will bring up the usual descriptive material.

## Reports

OpenBiblio has a very flexible report system.

### Running a report

1. Click on Reports.
2. Find the report you'd like to run and click on it.
3. OpenBiblio will ask you to add any additional necessary specifications.
4. You may sort the report by clicking on the headings.

### Exporting to different formats

You may export your report results in PDF, CSV, or MARC formats. If you'd like to print the results, we recommend the PDF format (found after you run a report under Reports → Report Results → Print List).

### Adding new reports

You need access to the server via FTP, RDP, or SSH.

## Managing OpenBiblio Plugins

OpenBiblio has a very flexible plugin system. This document will describe how you can enable and disable plugins, understand the plugins that are included with OpenBiblio, and design a new plugin.

### Enabling plugins

---

**Tip**

You will need access to the Tools menu.

---

1. Go to Tools → Plugin Manager.
  2. Make sure that Plugins are Allowed.
  3. Check the box next to each plugin you are interested in using.
-

## Plugins bundled with OpenBiblio

These plugins all come bundled with a default OpenBiblio installation.

### CSS Utilities (*cssUtils*)

This plug-in will look for two categories of CSS issues:

1. CSS entries in `.../styles.css` that appear to not be in use
2. CSS classes referenced in OB that do not appear in `.../styles.css`

### WARNINGS

No fixes are suggested or made. The user should be VERY careful in removing any that appear to not be in use as some css usage is dynamic via jQuery and is difficult to detect programmatically.

Query selectors having multiple entries (e.g. `$(.reqd sup)`) are not currently parsed as multiple entries to be searched.

### Pull-Down List Manager (*List Manager*)

This plug-in is intended to provide a developer with a simple means to view the contents of pull down lists that are used in various locations throughout OB.

No means are provided at this time to modify these lists; use phpMyAdmin or equiv for that purpose.

### Media Fields

This plug-in is intended to provide a means for OB users to exchange input form layouts.

Some media forms are infrequently used and a new user may not know just what material should be collected.

Using this plugin, it is possible to create a text file of a layout that can be emailed to another for them to import.

### Orphan File finder (*orpahnFiles*)

This plug-in is intended to find old abandoned files. The concept here is to find files which:

1. are not referred to via any menu
2. are not referred to by any HTML mechanism such as Javascript, CSS, Form Actions, Images, etc.
3. are not referred to by any PHP mechanism such as `'include's` or `'require's`
4. are not referenced in Object class inheritances.

---

#### Note

there are many legacy files and folders that have an initial letter 'x'. These are to be considered *Legacy* works and may be removed from the OB project at the Lead Developer's discretion.

---

---

#### Note

There is an entire `.../docs` folder that has never been integrated into OB ver 1.0. this folder may be removed from the OB project at the Lead Developer's discretion.

---

### Translation Utilities (transUtils)

This plugin is intended as an aid to the creation or maintenance of a locale translation. All T(...) entries in OB .php files are checked. Each available translation is tested separately.

The following functions are available:

1. Check for Duplicate Entries. Intended to find duplicate entries which may be out of place alphabetically and so un-noticed.
2. Check for Unused Entries. Look for table entries which have been abandoned, or through miss-spelling somewhere, are not currently being used.
3. Check for Needed Entries. Look for T(...) entries which do not have a corresponding entry in the trans.php file.
4. Check for Potential Entries. Look for quoted strings (both normal display, and error) in all files which do not have a T(...) surround. Error messages embedded in the various language processors (PHP, JS, CSS, mySQL, etc) will not be caught.

### Call Number Utilities (CallNoUtils)

This section describes the Call Number Utilities plugin.

A pull-down allows a search for:

1. items without an assigned call number
2. proposed call numbers based on a desired schema
3. posting proposed call numbers as needed.

### Find Records without call numbers

This function will scan the entire database and return a list of all biblios which do not have a call number assigned.

### Dry Run of call number add

This function will propose call numbers based on the preferred source.

### Add call numbers to records

This function will post the proposed call numbers.

### Online photos (OpenLibrary)

This plugin takes cover images from the OpenLibrary project.

## Creating a new plugin

This requires basic PHP skills and access to the files.

### Design the plugin

1. Create a new directory that begins with the word plugin, followed by an underscore, followed by the name of your plugin. In UNIX systems, you might type something like this:

```
mkdir plugin_excitingPlugin
```

2. If you need to include custom CSS or JS, add it to a file called custom\_head.php in this new directory.
  3. Create three PHP files in the directory: a JS file, a Srvr file, and a Forms file.
-



## Create a plugin form

Your users will interact with your plugin using a form, located at `plugin_excitingPlugin/excitingForms.php`.

A basic plugin consists of a require statement to get the necessary methods, some code establishing basic UI, some HTML form markup, and a section to display results and messages. Here's a simple example:

```
<?php
    require_once("../shared/common.php");

    $tab = "tools";
    $nav = "ExcitingMgr";

    require_once(REL(__FILE__, "../shared/logincheck.php"));
    Page::header(array('nav'=>$tab.'/'.$nav, 'title'=>''));

?>
<h1 id="pageHdr" class="title"><?php echo T("excitingMgr"); ?></h1>
    <section>
        <fieldset id="exciting">
            <label for="site_cd" >Choose your favorite library:<select id="site_cd" ></
                select></label>
            <input type="button" id="showResultsBtn" value="Show results" />
        </fieldset>
    </section>
    <div id="rslts"></div>

<?php
require_once(REL(__FILE__, "excitingJs.php"));
?>
```

## Create a plugin JS file

Your plugin's JS file—located at `plugin_excitingPlugin/excitingJs.php`—is responsible for filling your Form with dynamic data, whether it is filling drop-down menus or presenting the results of your query from the server.

A basic JS file consists of functions that capture data from the user and send it to the server file(s). Here's a simple example:

```
<script language="JavaScript" defer>
"use strict";

var excite = {
    init: function () {
        excite.url = 'excitingSrvr.php';
        excite.listSrvr = "../shared/listSrvr.php";
        excite.fetchSiteList();

        $('#showResultsBtn').on('click', null, excite.doSendUserData);
    },
    fetchSiteList: function () {
        $.getJSON(excite.listSrvr, {mode:'getSiteList'}, function(data) {
            var html = '';
            for (var n in data) {
                html+= '<option value="'+n+' ">'+data[n]+'</option>';
            }
            $('#site_cd').html(html);
        });
    },
    doSendUserData: function () {
        $.post(excite.url, {'mode':$('#site_cd').val()},
            function (response) {
                $('#rslts').html(response);
            });
    }
};
```

```

                                $('#rslts').show();
                            });
                        },
                    };
$(document).ready(excite.init);
</script>

```

## Create a plugin server

Your plugin's server—located at `plugin_excitingPlugin/excitingSrvr.php`—is responsible for three tasks:

1. Performing any actions the user requests.
2. Obtaining any relevant data from the database, file structure, or external API.
3. Preparing a nice HTML display of the relevant data, which your JS file will add to the Form to display to your user.

A basic plugin server consists of a `require` statement to get the necessary data and methods and a `switch` statement to take care of the `$_POST['mode']` variable. Here's a simple example:

```

<?php
    require_once('../shared/common.php');
    switch ($_POST['mode']) {
        case 1:
            echo 'That\'s my favorite library too!';
            break;
        case 2:
            echo 'That library is pretty good.';
            break;
        default:
            echo T('invalid mode');
            break;
    }

```

---

### Note

If you would like to follow OpenBiblio style and serve up JSON from your Server to be processed by your JS file, be sure to use POST requests, rather than GET requests, to avoid the potential of Javascript Hijacking.

---

If you would like to access data from the database, you should require the appropriate model from the model directory. For example, if you'd like to use the `Biblios` model, you would include this in your `plugin_excitingPlugin/excitingSrvr.php`:

```

require_once('../model/Biblios.php');
$bib = new Biblios;

```

You can then access all those great methods and data through the `$bib` object.

## Add your plugin to the OpenBiblio menu

1. Create a new directory inside the plugins directory with the name of your plugin. In UNIX systems, you might type something like this:

```
mkdir plugins/excitingPlugin
```

---

2. Create a file within this new directory called nav.nav

```
<?php
Nav::node('tools/excitingPlugin', 'Exciting Plugin', '../plugin_excitingPlugin/ ↵
    excitingForms.php');
```

3. If you'd like to use Obib's localization framework but need to localize some more terms, create a file called tran.tran. You can use this to add new terms to your locale.

---

**Tip**

Plugins already have all of the exciting localization ability built in. However, note that you cannot override localized term definitions within tran.tran.

---

**Warning**

There is not yet a place to specify localized terms for plugins.

---

4. Turn on your plugin using Tools → Plugin Manager.

## OpenBiblio Recipes

This is a collection of neat tricks and configurations that might be nice for your library.

### A successful initial configuration

1. Go to Admin → Library settings. Set the character set to *utf8*.

### Adding a link to your system documentation

You can configure the "Help" link in OpenBiblio's menu to link to any URL you would like. This defaults to the official OpenBiblio documentation, but you may wish to replace it with some help that is customized for users at your library.

---

**Note**

You will need access to the SQL database to do this; it has not yet been added to the end-user interface.

---

1. Open up the SQL database using the interface of your choice (be it PhpMyAdmin, a command-line interface, or any other tool you prefer).
  2. Type `UPDATE settings SET value='https://myLibraryDocs.example.com' WHERE name='help_link'` (using your own URL instead of `https://myLibraryDocs.example.com`).
  3. Refresh the OpenBiblio interface in your browser. When you click on the help link, it should pop up a new window with your custom documentation.
-

## Changing the timeout length on HMAC requests

OpenBiblio encrypts certain requests that might expose confidential member data. Every time a user opens or refreshes a \*Forms.php page that might expose such data, the clock starts ticking and they must make their requests before a certain number of minutes elapse. This is to make sure that if the request falls into the wrong hands, the intruder will not have access to these data for very long.

You can set the number of minutes before sensitive requests time out to the value that works best for your library, based on factors like your workflow and security concerns.

1. Open up the SQL database using the interface of your choice (be it PhpMyAdmin, a command-line interface, or any other tool you prefer).
2. Type `UPDATE settings SET value='5' WHERE name='hmac_timeout'` where 5 is the number of minutes before sensitive requests become invalid.

---

**Tip**

If you would like to "turn off" this timeout feature (technically setting the expire time to 28 days), set the value to 0.

---

## Accessibility

### Skiping navigation menus

If you are using OpenBiblio with screenreader software, you may not be interested in listening to your software recite all of OpenBiblio's navigation links every time you move to a new page. You can instead select the link called "Skip to main content," which will take you to the content that you are more interested, such as the catalog or circulation interface.

You may also access this link using your keyboard using Tab key repeatedly.

---